

REMARKS/ARGUMENTS**I. General**

Claims 1-20 were pending in the current application. A Final Office Action was mailed September 30, 2002 in which all of claims 1-20 were rejected. The outstanding issues raised in the Final Office Action of September 30, 2002 are:

- Claims 1-5, 11-13, 16 and 18-20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over alleged applicant admitted prior art; and
- Claims 6-10, 14-15 and 17 are rejected under 35 U.S.C. § 103(a) as being unpatentable over alleged applicant admitted prior art in view of *SQL ServerälTransact-SQL User's Guide*.

In response to such Final Office Action, Applicant files a Continuing Prosecution Application (CPA) with this accompanying preliminary amendment, in which Applicant amends claims 1, 13, and 18, deletes claims 15 and 19 without prejudice, and adds new claims 21 and 22. Applicant respectfully traverses the outstanding claim rejections, and requests reconsideration and withdrawal in light of the amendments and remarks presented herein.

II. Claim Amendments

Applicant sets forth amendments to claims 1, 13, and 18, deletes claims 15 and 19 without prejudice, and adds new claims 21 and 22 herein. Accordingly, claims 1-14, 16-18, and 20-22 are now pending in this Application. No new matter has been added by the amendments and claim additions presented herein.

More specifically, independent claim 1 is amended herein to recite “using said query for monitoring said system for existence of said condition of said attribute” in place of the previously recited limitation of “querying said system as specified by said request”.

Independent claim 13 is amended herein to correct a typographical error by changing an occurrence of “attributes” to “attribute”. This amendment to claim 13 is intended merely as a cosmetic change and is not intended to narrow the scope of claim 13 in any manner.

Independent claim 18 is amended herein to recite “application program” after each occurrence of “client” therein.

III. Response to Claim Rejections

The claims of the present application were rejected in the Final Office Action in the manner identified above. Applicant respectfully submits that the claims, as amended herein, are allowable over the applied references of record, as described more fully below.

Independent Claim 1

For example, independent claim 1, as amended herein, recites, *inter alia*, “receiving a request from a client to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute; using said query for monitoring said system for existence of said condition of said attribute . . . and upon determining that said condition exists, notifying said client of the existence of said condition” (emphasis added). The prior art described in the present application fails to provide at least this limitation of claim 1.

The applied art (i.e., that described in the present application) fails to teach or suggest receiving a request from a client to notify the client of a condition of an attribute of a system, monitoring the system for the existence of the condition, and notifying the client of the existence of the condition upon determining that the condition exists. Rather, in the prior art described in the present application, an application program itself may periodically query a system and determine whether a condition of an attribute exists at that time. Of course, this requires that the application program itself monitor the system for the existence of a condition of an attribute through periodic querying of the system. The prior art fails to teach or suggest receiving a request specifying a query for a system attribute from a client, monitoring the system for the existence of the condition, and notifying the client of the existence of the condition upon determining that the condition exists.

Independent Claim 18

As another example, independent claim 18, as amended herein, recites, *inter alia*, “a means for executing said reporting application; wherein said reporting application includes

computer executable software code for receiving from a client application program a request to notify said client application program of a condition of an attribute of a system, said request comprising information specifying a query for said system attribute, computer executable software code for determining if said condition exists, and computer executable software code that, upon determining that said condition exists, notifies said client application program of the existence of said condition" (emphasis added). The prior art described in the present application fails to provide at least this limitation of claim 18.

The applied art (i.e., that described in the present application) fails to teach or suggest computer executable software code for receiving from a client application program a request to notify the client application program of a condition of an attribute of a system. Further, the applied art fails to teach or suggest computer executable software code that, upon determining that the condition exists, notifies the client application program of the existence of the condition. Rather, in the prior art described in the present application, an application program itself may periodically query a system and determine whether a condition of an attribute exists at that time.

Independent Claim 13

As for independent claim 13, Applicant reasserts the arguments presented in the amendment mailed July 24, 2002, and requests that the Examiner reconsider those arguments in view of the further comments provided herein below.

Claim 13 is directed to a reporting application for reporting the existence of a specified condition in a system attribute to a client, and recites, *inter alia*, "computer executable software code for receiving from a client a request to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute; . . . computer executable software code for deriving data about said system attribute; . . . and computer executable software code that, upon determining that said condition exists, notifies said client of the existence of said condition" (emphasis added). The prior art described in the present application fails to provide at least this limitation of claim 13.

The applied art (i.e., that described in the present application) fails to teach or suggest a reporting application that comprises computer executable software code for receiving from a client a request to notify the client of a condition of an attribute of a system. Further, the applied art fails to teach or suggest a reporting application that comprises computer executable software code that, upon determining that the condition exists, notifies the client of the existence of the condition.

Response to the Examiner's Remarks in the Final Office Action for Claim 13

The Examiner appears to apply two separate implementations of the prior art that are described in the present application in rejecting claim 13. The two separate implementations of the prior art that are described in the present application are summarized as follows:

First, in the prior art an application program may itself operate to issue commands querying a system, receive actual data responsive to the query, and derive information about system attributes. For example, page 2, lines 1-5 of the present application states in part:

“...a prior art application program may operate to investigate and obtain information about system attributes, and then such program may itself figure out whether any changes have occurred...”

Similarly, page 3, lines 21-23 of the present application states, “...in the prior art, an application program can issue commands querying the system, and in response to such commands receive ‘actual’ data.”

From the above prior art, an application program can issue commands querying the system and deriving information about system attributes for determining itself whether a particular condition exists. For example, supposing that an application program needs at least 4 nodes to be present in a cluster for the application program to execute (or to execute in a particular manner), in the prior art, the application program itself may query the system for its nodes, and from the actual data received from the query the application program may itself determine whether at least 4 nodes are present.

The application program may use a system call to an operating system to query the system. For instance, page 3, lines 3-13 of the present application provides:

More specifically, applications can execute a command (or series of commands) to query the cluster in which the application is executing or some external cluster to determine the membership of such cluster. For example, within the UNIX operating system, an application program may utilize a command to obtain the current status of the cluster (e.g., whether the cluster is up or down) and the membership of such cluster. Alternatively, an application program may utilize its own command to obtain information regarding membership of the cluster, instead of utilizing a system call to use the operating system's command.

In this first prior art implementation, the application program does not have a "client" from which it receives a request to be notified of a condition of an attribute of a system. Further, the application program does not have a "client" to which it notifies of the existence of a condition upon determining that the condition exists.

The Examiner asserts in the Final Office Action that "[b]ecause the application could not automatically issue the command and in order to issue the commands by an application program, obviously, a user, who gives the commands or requests to the application program is present." Page 3 of Final Office Action. Thus, the Examiner appears to contend that a client (e.g., a user) must necessarily be present in the prior art implementation. Applicant respectfully submits that the Examiner is incorrect in this regard, as described further below.

The application program need not necessarily have a client (e.g., a user), and certainly in this first prior art implementation described in the present application, the application program does not have a client. Rather, the application program is implemented in a manner such that it may query the system and determine whether certain conditions exist in certain attributes of the system that may effect the application program's execution. The application program performs the query as part of its implemented operation, rather than in response to a user-input request. For instance, before performing an operation that requires at least 4 nodes in a cluster, the application program may be implemented to query the system and determine whether at least 4 nodes exist. A request for whether this condition exists is not made by a client of the application program (e.g., user), but instead the application program makes the query and determination as part of its implementation. Thus, the application program is implemented such that it automatically performs the appropriate queries of the system to ensure that the system attributes are acceptable for performing a given operation without a user submitting the commands for performing the queries to the application program. Accordingly, in this first implementation of the prior art described in the present application,

an application program does not have a “client” that is requesting notification of a condition of an attribute of a system.

The second prior art implementation described in the present application on which the Examiner relies is one in which a paper or user is asynchronously notified of changes in system attributes. For example, page 5, line 28 - page 6, line 4 of the present application provides:

In the prior art, an application or user may be notified asynchronously of changes in system attributes. For example, a user may be notified if a printer on the system is out of ink, has a paper jam, or is out of paper. However, as explained above, if an application or user desires derived data, such as whether all printers existing on a network or within a cluster are out of paper, the user or application itself is required to derive such data from actual data received from querying the network or cluster.

From the above prior art, an application program or user may receive notification of changes in system attributes asynchronously. However, as with the first prior art implementation described above, a request for notification of the existence of a condition of an attribute of the system is not received from a client. For example, in this instance, a user or application program does not submit a request for notification of the existence of a change in system attributes, but rather the user or application asynchronously receives notification of a change for which the system provides notification (e.g., printer is out of ink).

Further, claim 13 is directed to a reporting application that comprises, *inter alia*, “computer executable software code for deriving data about said system attribute; computer executable software code for determining from said derived data if said condition exists; and computer executable software code that, upon determining that said condition exists, notifies said client of the existence of said condition” (emphasis added). As described in the above-quoted portion of the present application, in this second implementation of the prior art a reporting application is not provided for deriving data, determining from the derived data whether a condition exists, and notifying the client of the existence of the condition upon determining that it exists.

The Examiner appears to combine the two separate prior art implementations described above in rejecting claim 13 in the Final Office Action under 35 U.S.C. § 103(a). Even if sufficient motivation existed for combining the two implementations in the manner

proposed by the Examiner (which Applicant submits does not), the resulting combination fails to provide all of the limitations of claim 13. For instance, neither the first prior art implementation nor the second prior art implementation provide a reporting application that includes computer executable software code for receiving from a client a request to notify the client of a condition of an attribute of a system. As described above, the first prior art implementation provides an application program that does not have a "client" from which it receives a request for notification of a condition of an attribute of a system. Further, the second prior art implementation provides a system in which a user or application program is asynchronously notified of a change in a system attribute without the user or application requesting such notification. Thus, neither implementation provides a reporting application that includes computer executable software code for receiving from a client a request to notify the client of a condition of an attribute of a system. Therefore, the combination of the two implementations fails to teach or suggest each and every limitation of claim 13.

Further, no motivation exists for combining the first and second prior art implementations. In the Final Office Action, the Examiner asserts that "the motivation is to have a method for notifying the system attributes when a user would like to inquiry the system and have an appropriate feedback of changes or system attributes relate to the user's query". Page 5 of Final Office Action. The language of the recited motivation is circular in nature, stating that it is obvious to make the modification because it is obvious to achieve the result.

Further, it is not at all clear that the stated motivation would actually lead one of ordinary skill in the art to combine the first and second prior art implementations. For instance, the first implementation provides for an application program to query a system and derive needed information. The second implementation provides for a system asynchronously notifying an application (or user) of a change in a system attribute. It is not at all clear that if motivation exists for a user to query the system and receive feedback of changes in system attributes, such motivation would lead to the combination of the first implementation (in which an application program queries a system) and the second implementation (in which an application/user can asynchronously receive notification of a change of system attributes without a query being made). In neither implementation does a user query the system, and thus one motivated to enable a user to query a system would likely

not look to combining the first and second prior art implementations described above.

Additionally, the mere fact that references can be combined or modified does not render the resultant combination or modification obvious unless the prior art also suggests the desirability of the combination or modification. *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990), as cited in M.P.E.P. § 2143.01. The desirability of the combination proposed by the Examiner is not at all clear, and such desirability is certainly not suggested in the applied prior art (i.e., the description of the prior art in the present application). Thus, the motivation provided by the Examiner is improper, as the cited prior art reference must establish the desirability for making the modification.

Further still, the motivation supplied in the Office Action is apparently derived from Applicant's disclosure. See page 7, lines 1-24 of the present application. The teaching or suggestion to make the claimed combination must be found in the prior art, not in Applicant's disclosure, see M.P.E.P. § 2143, citing *In re Vaeck*, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991). As stated by the Federal Circuit:

It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. This court has previously stated that "[o]ne cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention."

In re Fritch, 972 F.2d 1260, 23 U.S.P.Q.2d 1780, 1784 (Fed. Cir. 1992) (quoting *In re Fine*, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1600 (Fed. Cir. 1988)).

Hindsight is almost always perfect. It is insufficient to prove that at the time of the claimed invention, the separate elements of the device were present in the known art. Rather, there must have been some explicit teaching or suggestion in the art to motivate one of even ordinary skill to combine such elements so as to create the same invention. See *Arkie Lures, Inc. v. Gene Larew Tackle, Inc.*, 119 F.3d 953, 957, 43 U.S.P.Q.2d 1294 (Fed. Cir. 1997). Such a teaching or suggestion is absent from the art applied by the Examiner. Thus, the motivation provided by the Examiner is improper, as the motivation must be described in a prior art reference and must detail the benefits of such a modification.

Dependent Claims

Dependent claims 2-12, 14, 16-17, and 20-22 each depend either directly or indirectly from one of base claims 1, 13, and 18, and thus inherit all of the limitations of their respective base claims. It is respectfully submitted that dependent claims 2-12, 14, 16-17, and 20-22 are allowable not only because of their dependencies from their respective base claims 1, 13, and 18 for the reasons discussed above, but also in view of their novel claim features (which both narrow the scope of the particular claims and compel a broader interpretation of the base claims from which they depend).

As an example, dependent claim 21 recites:

The reporting application of claim 13 wherein said computer executable code for querying said system comprises computer executable code for periodically querying said system. (Emphasis Added).

The applied prior art fails to teach or suggest this further limitation of claim 13 in which a reporting application comprises code for periodically querying a system.

CONCLUSION

Claims 1-14, 16-18, and 20-22 are now pending in the current application. As shown above, there are important differences between the claims and the applied art. Moreover, a person of ordinary skill in the art considering the prior art would not find these differences obvious. Accordingly, Applicant respectfully asserts that claims 1-14, 16-18, and 20-22 are allowable over the applied art. Therefore, Applicant respectfully requests that these claims be passed to issue.

Applicant respectfully requests that the Examiner call the below listed attorney if the Examiner believes that such a discussion would be helpful in resolving any remaining problems.

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" service under 31 C.F.R. 1.10 on the date indicated below and in an envelope addressed to: Commissioner for Patents, Washington, D.C., 20231.

Date of Deposit: November 14, 2002

Typed Name: Kristin Mattern

Signature: KMattern

Respectfully submitted,

By 

Jody C. Bishop

Attorney/Agent for Applicant(s)

Reg. No.: 44,034

Date: November 14, 2002

Telephone No. (214) 855-8007

Version With Markings to Show Changes Made

1. (Amended) A method of reporting existence of a specified condition in a system attribute, said method comprising:

receiving a request from a client to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute;

[querying] using said query for monitoring said system [as specified by said request;]
for existence of said condition of said attribute;

deriving data about said system attribute to determine if said condition exists; and
upon determining that said condition exists, notifying said client of the existence of said condition.

13. (Amended) A reporting application for reporting the existence of a specified condition in a system attribute to a client, said reporting application comprising:

computer executable software code for receiving from a client a request to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute;

computer executable code for querying said system as specified by said request;

computer executable software code for deriving data about said system [attributes;]
attribute;

computer executable software code for determining from said derived data if said condition exists; and

computer executable software code that, upon determining that said condition exists, notifies said client of the existence of said condition.

15. Please delete claim 15 without prejudice.

18. (Amended) A system for reporting the existence of a specified condition in a system attribute to a client, said system comprising:

means for storing a reporting application;

a means for executing said reporting application;

wherein said reporting application includes computer executable software code for receiving from a client application program a request to notify said client application program of a condition of an attribute of a system, said request comprising information specifying a query for said system attribute, computer executable software code for determining if said condition exists, and computer executable software code that, upon determining that said condition exists, notifies said client application program of the existence of said condition.

19. Please delete claim 19 without prejudice.

Please add new claims 21 and 22 as follows:

21. (New) The reporting application of claim 13 wherein said computer executable code for querying said system comprises computer executable code for periodically querying said system.

22. (New) The system of claim 18 wherein said reporting application further comprises:

computer executable software code for monitoring said system to determine if said condition exists.